RUTGERS UNIVERSITY
Center for Expert Systems Research

Quarterly Report:
Empirical Analysis and Refinement of Expert System Knowledge Bases
Contract Number
N00014-87-K-0398
Office of Naval Research

August 31, 1989

Principal Investigators:
Sholom M. Weiss
Casimir A. Kulikowski

Table of Contents

1. Technical Project Summary

Knowledge base refinement is the modification of an existing expert system knowledge base with the goals of localizing specific weaknesses in a knowledge base and improving an expert system's performance. Systems that automate some aspects of knowledge base refinement can have a significant impact on the related problems of knowledge base acquisition, maintenance, verification, and learning from experience. The SEEK system was the first expert system framework to integrate large-scale performance information into all phases of knowledge base development and to provide automatic information about rule refinement. A recently developed successor system, SEEK2 [Ginsberg, Weiss, and Politakis 88] significantly expands the scope of the original system in terms of generality and automated capabilities. The investigators expect to make significant progress in automating empirical expert system techniques for knowledge acquisition, knowledge base refinement, maintenance, and verification.

2. Principal Expected Innovations

The investigators will demonstrate a rule refinement system in an application of the diagnosis of complex equipment failure: computer network troubleshooting. The expert system should demonstrate the following advanced capabilities:

- automatic localization of knowledge base weaknesses
- automatic repair (refinement) of poorly performing rules
- automatic verification of new knowledge base rules
- automatic learning capabilities

3. Objectives for FY89

These are our objectives for the current year, Fiscal year 89:

- full demonstration of refinement system, using subset of DEC's Network Troubleshooting Consultant (NTC). System will automatically recover from many forms of damage to knowledge base.
- full demonstration of system with capabilities for automatic refinement, and verification of knowledge base consistency. Empirical experiments will be performed and results will be reported.
- demonstration of significant automated rule learning capabilities.
- demonstration of extended system capabilities for alternative control strategies and representations.

- completed comparative studies of empirical techniques for machine learning, statistical pattern recognition, and neural nets.

4. Summary of Progress

During the previous year the following was accomplished:

- initial functioning equipment diagnosis and repair knowledge base, suitable for refinement. This is a subset of DEC's Network Troubleshooting Consultant (NTC).

- initial demonstration of functioning equipment diagnostic system with capabilities of localization of weak rules, automatic refinement, automatic verification.

- demonstration of initial rule learning capabilities.

- development of case generation simulator and randomized rule modifier.

- initial comparative studies demonstrating superiority of PVM rule induction procedure in low dimensional applications.

This work is the basis for further progress in developing an automated refinement system. We are pursuing the refinement and learning tasks from both an expert system rule-based perspective and a machine learning rule induction perspective. In order to develop the strongest form of refinement system, we have examined numerous techniques for empirical rule induction. We have also developed a procedure, Predictive Value Maximization [Weiss, Galen, and Tadepalli 90], that shows strong results for induction of single relatively short rules. Our fundamental objective is to mix the best rule induction procedures with a rule-based expert system to achieve the strongest empirical results.

Here are the highlights of new progress in meeting our stated objectives for fiscal year 89:[1]

- We have completed an extensive empirical comparison of machine learning rule induction techniques with statistical pattern recognition techniques, and neural nets. Four real-world data sets were analyzed using different techniques. The study required over 6 months of Sun 4 CPU time. The results are described in a completed paper that was presented at the 1989 International Joint Conference on Artificial Intelligence [Weiss and Kapouleas 89].

- We have completed a procedure for the refinement system that uses rule induction techniques. This procedures gives the refinement system a *learning* capability which is the most difficult and important of our major research objectives for this fiscal year.

[1]We have received a no-cost extension of our contract to the end of calendar year 1989. A final report will be issued at that time.

The fundamental approach of rule refinement is to constrain changes that can be made to the knowledge base to those that are fully consistent with the rules of the expert-supplied knowledge base. Unlike a refinement system, a pure learning system such as a rule induction system, attempts to learn directly from data, unconstrained by human expert knowledge. A more constrained learning approach maintains the expert supplied rules but allows for some additions to the rules. The new learning procedures added to the refinement system use generalization and specialization models to perform 2 functions:

- add a variable to a rule to specialize the rule
- add a new rule to the knowledge base to generalize the rule

The procedure for adding components and rules is detailed in Section 4.1. Some key parts of the procedure are analogous to current tree generation procedures such as ID3/C4 or CART, where the split is performed on the single best node. In our case during a given refinement cycle, we attempt to induce the single best variable and decision threshold. The following preliminary results were found for a knowledge base of 100 rules and 5 endpoints that previously was refined from a performance of 73% (88/121) to 100% (121/121).

- The same 100% refinement performance was achieved with the learning capability.
- When all 100 rules, with an average of 4 variables per rule, were deleted from the knowledge base, the system was able to generate 14 rules and 21 variables that achieved 88% (107/121) correct classification.

As a pure learning procedure, these techniques are somewhat weaker than induced decision trees. The heuristic refinement strategy of generalization and refinement does not appear to perform as well when train and test simulations are used to estimate the true error rate. However, this refinement strategy is not meant to be a learning strategy that applied only to sample data. It can readily work on an existing knowledge base and produces a new knowledge base that is consistent with the original expert derived knowledge base. These results demonstrate the potential for robust mixed knowledge base refinement and learning procedures.

Additional results for learning with the Network Troubleshooting Consultant are listed in table 4-1. In these simulations, the knowledge base was perturbed, and then the refinement system attempted to *fix* the knowledge base. Each *bash* is one random modification to a rule attribute in the knowledge base. Table 4-1 lists the number of random changes made to rules in the knowledge base, the subsequent performance of the rule-bases system using these bashed rules as measured in correct cases, the number of refinements the leaning system makes to the knowledge base, and the subsequent performance after refinement. There are 74 stored cases.

In addition to the learning techniques, a limited language was developed for constraining the

no. of bashes	correct cases	num. of refinements	refined correct
1	74	-	-
2	74	-	-
4	74	-	-
8	72	1	74
16	69	4	74
32	66	3	74
64	66	2	74
128	57	5	72
256	47	6	72

Figure 4-1: Refinement of Randomly Perturbed Knowledge Base

refinement process based on domain specific characteristics. The following constraints were implemented and tested:

- Disallow modifications to a specified set of rules.
- Disallow any refinements that reach erroneous conclusions for any case in a set of specified cases.
- Restrict learning refinement such that only attributes from the specified set may be used to add to an existing rule or to form a new rule.

4.1. Refinement Learning Procedure

The following procedure briefly outlines the techniques used to add components to existing rules and to create new rules:

Add a Finding to a rule: Specializing the Rule

1. While calculating the statistics for use by the heuristics, store a list of GAIN and LOSS cases for each rule. GAIN is the number of cases that would be gained if the rule was eliminated. LOSS is the number of cases that would be lost if the rule was eliminated.

2. The requirement for trying an experiment is that GAIN(rule)>0. Probable gain is less than or equal to GAIN.

3. Mark the LOSS cases as H+, the GAINs as H-, others ignored.

4. Generate the best attribute to be added to this rule.

5. If there is a best attribute, add it to the rule under consideration. Test.

Add A New Rule To The Knowledge Base:Generalization

1. Calculate the number of false positive and false negative cases for a given conclusion. If there are more FPs than FNs, skip the heuristic. Else proceed.

2. Go through all the cases. Mark all unknown, test cases and true positive cases to be ignored. Mark the FN cases as H+, and the rest as H-.

3. Generate the best attribute to be used as a new rule.

Generating the Best Attribute

The following table is computed for each attribute over the indicated set of cases:

	Attribute true	Attribute false
H+ cases	A	B
H- cases	C	D

1. Loop through the true/false findings. For each attribute FIN, consider both true and false attributes. Loop through each case to set up a predictive analysis table for each attribute.

2. Calculate the estimators and probable gain for each attribute.

   - a. For adding to an existing rule, estimator = A+D-B-C probable gain = D-B

   - b. For a new rule, estimator = A+D-B-C probable gain = A

3. Save the attribute with the highest estimator.

4. Loop through each numerical finding FIN.

5. Loop through the H+ cases to get each numerical VAL. Consider each attribute at each cutoff with greater and less than operators. Loop through each case to set up a predictive analysis table for each attribute. Calculate the estimator for each attribute. Save the best overall.

6. If the probable gain>0, return the best attribute.

5. Financial Review

1. Basic contract dollar amount: $536,919 (9/1/87-12/31/89)

2. Dollar amounts and purposes of options: None

3. Total spending authority received to date: $475,000 through 1/31/89

4. Total spending to date: $439,697 through 8/31/89

5. Monthly expenditure rate: As anticipated, funding of larger portions of the summer salaries of the principal investigators over the past summer, as well as more systems programmer support, was provided. This was due to our increased efforts devoted to the research project during the summer as shown by the acceptance of a paper that was presented at the IJCAI-89, held in Detroit in August, and another paper accepted for publication in the AI Journal. The continuation of an additional graduate assistant to assist in this research, resulted in higher salary expenditures as anticipated for the 1988-89 academic year and summer of 1989.

6. We have expended a total of approximately $439,697 to date. This would, therefore, result in an average monthly expenditure rate of $17,588.

7. Major non-salary expenditures planned within this increment of funding: None

8. Date next increment of funds is needed: Immediately.

9. NOTE: The current expenditures, although approximate, now approach the total spending authority received to date. The spending authority has NOT been adjusted since January 1989 although a no-cost extension of the grant period was approved on July 10, 1989. We must have the spending authority adjusted to the full basic contract dollar amount to cover our expected expenditures through December 31, 1989.

References

[Ginsberg, Weiss, and Politakis 88]
Ginsberg, A., Weiss, S., and Politakis, P.
Automatic Knowledge Base Refinement for Classification Systems.
Artificial Intelligence :197-226, 1988.

[Weiss and Kapouleas 89]
Weiss, S., Kapouleas, I.
An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods.
In *International Joint Conference on Artificial Intelligence*, pages in press. Detroit, Michigan, 1989.

[Weiss, Galen, and Tadepalli 90]
Weiss, S., Galen, R., and Tadepalli, P.
Maximizing the Predictive Value of Production Rules.
Artificial Intelligence :in press, 1990.